正则表达式及其应用

Walter Zhou

2005-9

内容简介

- 基本的元字符(Meta-Character)介绍
- 正则表达式匹配的两大原则
- 括号（Parentheses）在正则表达式的应用
- 正则表达式在Tools中的应用
- 正则表达式在Programming Language中的应用
- Q&A

Note: about 90分钟

正则表达式能干什么？

zl_dragon@163.com

Walter.zhou@chn.xerox.com

Test@13.187.78.12

…

怎样检查email格式的合法性？

正则表达式能干什么？

姓名	生日	性别	进公司
AAA	1980-1-1	male	2004-1-1
BBB	1982-4-1	female	2005-4-1
CCC	1976-8-9	male	2002-5-6

查生日与进公司同一天的员工？

正则表达式能干这个！

zl_dragon@163.com

Walter.zhou@chn.xerox.com

Test@13.187.78.12

…

怎样检查email格式的合法性？

^([\w-\.]+)@((([0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.)|(([\w-]+\.)+))([a-zA-Z]{2,4}|[0-9]{1,3})$

正则表达式能干这个！

姓名	*生日*	*性别*	*进公司*
AAA	*1980-1-1*	*male*	*2004-1-1*
BBB	*1982-4-1*	*female*	*2005-4-1*
CCC	*1976-8-9*	*male*	*2002-5-6*

查生日与进公司同一天的员工？

([A-Za-z]+[:space:]+[A-Za-z]+)\s+19\d\d-(\d{1,2})-(\d{1,2})\s+(male|female)\s+(19|20)(\d{1,2})-\2-\3

基本的元字符(Meta-Character)介绍

Metacharacter	Name	Meaning
.	Dot	any on character
[…]	character class	any character listed
[^…]	negated character class	any character not listed
^	caret	the position at the start of the line
$	dollar	the position at the end of the line
\<	backslash less-than	*the position at the the start of a word
\>	backslash greater-than	*the position at the the end of a word
|	or; bar	matches either expression it separates
(…)	parentheses	used to limit the scope
\d	digit	same as [0-9]
\D	non-digit	same as [^0-9]

扩展的基于POSIX的“character class”

- [:alnum:] alphabetic chars & num chars
- [:alpha:] alphabetic chars
- [:blank:] space & tab
- [:cntrl:] control chars
- [:digit:] digits
- [:graph:] non-blank(not spaces, control chars, or the like)
- [:lower:] lowercase alphabetic
- [:upper:] uppercase alphabetic
- [:print:] like [:graph:], but + space char
- [:space:] all whitespace chars ([:blank:], newline, carriage return, and the like)
- [:xdigit:] hexadecimal number

Quantifier --- Repetition Metacharacter

• ?	One optional
• *	any amount optional
• +	some required
• {min, max}	from min to max

Note:

? ⟷ {0,1}

** ⟷ {0,}*

+ ⟷ {1,}

简单的例子

- [0123456789] match any single digit
- [0-9] ditto
- [0-9\-] match 0-9, or minus
- [a-z0-9] match any single lowercase letter or digit
- [^0-9] match any single non-digit
- [^\^] match single char except an up-arrow
- [\da-fA-F] match one hex digit

简单的例子(Cont.)

- [Cc]olou?r Colour, colour, color, Color
- bar{3} barrr
- (bar){3} barbarbar
- \<the the, theory, another
- ^The The house …
- ^cat$ There is only one “cat” in a line
- X{0,0} meaningless,compare with [^X]
- ^$ empty line

优先级

- a|b* 表示什么？
  - 任意多个a或b？
  - 一个a和多个b？

优先级

名称	表达式
扩号	()
重复式	? + * {m,n}
顺序和定位符	abc ^ $ \W \D
或选式	|

优先级

- *a|b* 表示什么？*
  - *任意多个a或b？*
  - *一个a和多个b？*

- a|b* 表示什么？
  - 任意都个a或b？
  - 一个a和多个b

正则表达式匹配的两大原则

- *The Earliest match wins*
- *The Quantifiers are greedy*

The Earliest match wins

- 最早最优先或称左优先

match “cat”in the following string

The dragging belly indicates your cat is too fat

正确的应为“\<cat\>”

The Quantifiers are greedy

- ?,*,+, and {min, max} 是贪婪鬼,它们匹配尽可能“多”的模式

例子：

$string = “a xxx c xxxxxxxxx c xxx d”

*/a.*c.*d/*

(.) 代表 “a xxx c xxxxxxxxx c xxx d” 还是*

“a xxx c xxxxxxxxx c xxx d”

The Quantifiers are greedy

- ?,*,+, and {min, max} 匹配尽可能“少”的模式

例子：

$string = “a xxx c xxxxxxxxx c xxx d”

/a.?c.*d/*

(.?) 代表 “a xxx c xxxxxxxxx c xxx d”*

Note:

在任何quantifier的后面都可以有+?来匹配尽可能“少”

括号（Parentheses）在正则表达式的应用

括号除了区分正则表达式的优先级外，更加巧妙的一个功能是具有“记忆”

例子：

$string = “a xxx c xxxxxxxxx c xxx d”

/a（.）c（.*）d/*

The 1st (.*) --- $1---代表“a *xxx c xxxxxxxxx* c xxx d”

The 2nd (.*) --- $2 ---代表“a xxx c xxxxxxxxx c *xxx* d”

括号（Parentheses）在正则表达式的应用

例子:

<City>Shanghai</City>

<Country> China</Country>

<Birthday> 2005-12-31</Birthday>

<Name>John</Name>

……

请用一个正则表达式提取出各种TAG中的内容？

括号（Parentheses）在正则表达式的应用

例子:

<City>Shanghai</City>
<Country> China</Country>
<Birthday> 2005-12-31</Birthday>
<Name>John</Name>
……
请用一个正则表达式提取出各种TAG中的内容?

<(\w)>(.)</\1>*

$1 ➔ \w
*$2 ➔ .**

括号（Parentheses）在正则表达式的应用

例子:

查生日与进公司同一天的员工

姓名	生日	性别	进公司
AAA	1980-1-1	male	2004-1-1
BBB	1982-4-1	female	2005-4-1
CCC	1976-8-9	male	2002-5-6
……			

括号（Parentheses）在正则表达式的应用

例子:

查生日与进公司同一天的员工

姓名	*生日*	*性别*	*进公司*
AAA	*1980-1-1*	*male*	*2004-1-1*
BBB	*1982-4-1*	*female*	*2005-4-1*
CCC	*1976-8-9*	*male*	*2002-5-6*

……

([A-Za-z]+[:space:]+[A-Za-z]+)\s+19\d\d-(\d{1,2})-(\d{1,2})\s+(male|female)\s+(19|20)(\d{1,2})-\2-\3

姓名:	[A-Za-z]+[:space:]+[A-Za-z]+	$1
生日:	19\d\d-(\d{1,2})-(\d{1,2})	$2 and $3
性别:	male|female	$4
进公司:	(19|20)(\d{1,2})-\2-\3	$5 and $6

正则表达式在Tools中的应用

- sed (stream editor)
- AWK
- egrep
- vi
- emacs

……

egrep

- 搜索系统中完全以数字为目录的path

```
find / -type d -print | egrep "/\<[0-9]+\>/"
```

正则表达式在Programming Language中的应用

- Perl
- Javascript
- C++ (boost library)
- Python

……

几乎所有语言都支持正则表达式

正则表达式在Perl中的应用

- Match

```
Open (FILE, ‘c:/config/员工录’) || die “fail\n”
While($line = <FILE>)
{
    if(
      $line =~ /
                    ([A-Za-z]+[:space:]+[A-Za-z]+)\s+19\d\d-(\d{1,2})-
                    (\d{1,2})\s+(male|female)\s+(19|20)(\d{1,2})-\2-\3
                /
     )
    {
      print “生日 =进公司日期\n”
      print “Name: $1\n”;
      print “Sex: $4\n”;
      print “Birthday: $2-$3\n”
    }
}
```

正则表达式在Perl中的应用

Example:

“exasperate” =~ /e(.*)e/

$1 = xasperat

“exasperate” =~ /e(.*?)e/

$1 = xasp

“exasperate” =~ /.*e(.*?)e/

$1 = rat

正则表达式在Perl中的应用

```
$burglar = “Bilbo Baggins”;
while($ burglar = ~ /b/gi)
{
    printf “Found a B at %d\n”, pos($burglar)-1;
}
```

Result:

Found a B at 0

Found a B at 3

Found a B at 6

C++中对正则表达式的支持

C++语言本身不认识什么叫“正则表达式”
通过Library来支持。

推荐Boost库Regex support